

VARISYSTEMS CORPORATION
PAC-16/DISC CONTROLLER
(PAC-16/DC)

The PAC-16 Controller with proper interface is now offered for use as a random access bulk storage driver.

A movable head, magnetic disc storage device, such as the Control Data 9432, 9433, 9434, 9463 or IBM 2311, can be used for this application.

Varisystems' PAC-16 can be provided for use with a single disc or up to eight discs on a common bus configuration. Storage capacities and access time for the various discs are summarized on the attached table.

A typical disc is organized into tracks, surfaces and sectors. Each recording surface of the disc is accessed by a movable head. The head can be moved to any of 200 tracks (100 tracks for the 9432). Each track can be divided into 1, 2, 4, 8, or 16 sectors by the PAC-16 interface.

The ten recording surfaces of the disc pack are addressed by the movable head assemblies. The packs used with the discs are industry standard replaceable disc packs (CDC 849 or IBM 1316).

Data transfer between the PAC-16 and the disc is accomplished in a block mode through the PAC-16 memory. This permits the editing of a single byte in a data block by transferring the block of data to the PAC-16 memory. By operating in this manner, we reduce the storage requirements of the main computer.

The recording format of the PAC-16/DC is obtained through software programming. The customer can vary the format to his requirements. In addition, the generation of labels and identifying tags, as well as checkword bytes, can be assigned through programming of the PAC-16 Controller.

Such simple external instructions as seek, read and write to the controller accomplishes data transfer to the disc. The PAC-16 internal command level instruction to the disc is generated from these simple instructions. Additional internal functions are: check byte and tag verification.

The PAC-16 reduces the size of the gaps in the recording format, increasing the efficiency of the data packing.

Example of a 9432 Disc using 8 sectors per track and a two byte home address for each sector. A 396 byte data area (including check bytes, but excluding synchronizing bytes and gaps) is packed for each sector. There are 8000 sectors available with the 9432 and a useful data area of 3,168,000 bytes.

The logo for Varisystems, featuring a stylized, handwritten-style script of the word "Varisystems" preceded by a horizontal line that curves into the letter 'V'.

VARISYSTEMS CORPORATION
PAC-16/DISC CONTROLLER
(PAC-16/DC)

Features of the PAC-16 Disc Storage Interface:

- Tight Packing Density
 - Typically for 8 sectors per track
 - H.A. 2 bytes
 - Data 396
 - Check Bytes 2
 - Sync & Data Recovery Patterns
- For 8000 sectors 3,168,000 data bytes/disc
- Industry Standard Interchangeable Disc Packs
- Internal Generation of Check Bytes
- Internal Verification of Check Bytes
- Internal Verification of Home Address Labels
- Simplified External Commands to Read Data, Write Data,
 Read Label, Write Label
- Internally Chained Read, Write Commands to Home Address
 Equal

The Varisystems' Controller provides an efficient and inexpensive systems utilizing a bulk storage device involving a random access memory under software control.

The logo for Varisystems, featuring a stylized, handwritten-style script of the word "varisystems" with a horizontal line extending from the left and a checkmark-like flourish on the right.

VARISYSTEMS CORPORATION
PAC-16/DISC CONTROLLER
(PAC-16/DC)

The PAC-16 Controller with proper interface is now offered for use as a random access bulk storage driver.

A movable head, magnetic disc storage device, such as the Control Data 9432, 9433, 9434, 9463 or IBM 2311, can be used for this application.

Varisystems' PAC-16 can be provided for use with a single disc or up to eight discs on a common bus configuration. Storage capacities and access time for the various discs are summarized on the attached table.

A typical disc is organized into tracks, surfaces and sectors. Each recording surface of the disc is accessed by a movable head. The head can be moved to any of 200 tracks (100 tracks for the 9432). Each track can be divided into 1, 2, 4, 8, or 16 sectors by the PAC-16 interface.

The ten recording surfaces of the disc pack are addressed by the movable head assemblies. The packs used with the discs are industry standard replaceable disc packs (CDC 849 or IBM 1316).

Data transfer between the PAC-16 and the disc is accomplished in a block mode through the PAC-16 memory. This permits the editing of a single byte in a data block by transferring the block of data to the PAC-16 memory. By operating in this manner, we reduce the storage requirements of the main computer.

The recording format of the PAC-16/DC is obtained through software programming. The customer can vary the format to his requirements. In addition, the generation of labels and identifying tags, as well as checkword bytes, can be assigned through programming of the PAC-16 Controller.

Such simple external instructions as seek, read and write to the controller accomplishes data transfer to the disc. The PAC-16 internal command level instruction to the disc is generated from these simple instructions. Additional internal functions are: check byte and tag verification.

The PAC-16 reduces the size of the gaps in the recording format, increasing the efficiency of the data packing.

Example of a 9432 Disc using 8 sectors per track and a two byte home address for each sector. A 396 byte data area (including check bytes, but excluding synchronizing bytes and gaps) is packed for each sector. There are 8000 sectors available with the 9432 and a useful data area of 3,168,000 bytes.

The logo for Varisystems, featuring a stylized, handwritten-style script of the word "Varisystems" preceded by a horizontal line that tapers into the letter 'V'.

TABLE - SUMMARY OF DISC STORAGE DRIVE CHARACTERISTICS

	Capacity in bytes	Number of Tracks	Access Time in Milliseconds		
			Max.	Average	1 Track
Control Data 9432	3 666 000	100	150	75	30
Control Data 9433	7 250 000	200 +3 spares	135	85	24
Control Data 9434 or Control Data 9463 or IBM 2311	7 250 000	200 +3 spares	135	85	24

In addition, the following is common to all drivers

Bits per byte: 8

Processing speed: 156,250 bytes/second

Rotational Delay Maximum: 25 milliseconds

Average: 12.5 milliseconds

Usable Disc Surfaces: 10

INTRODUCTORY MANUAL

for

PAC-16 PROGRAMMABLE ADAPTABLE CONTROLLER



Manufactured by

Varisystems



INTRODUCTORY MANUAL
for
PAC-16 PROGRAMABLE ADAPTABLE
CONTROLLER

Manufactured by
VARISYSTEMS [®] Corporation
207 Newtown Rd.
Plainview, New York 11803



FOREWORD

This manual contains selected material from Instruction Manual for PAC-16 Programable Adaptable Controller, VARISYSTEMS [®] Corporation, for the purpose of providing introductory information about the controller.

The Instruction Manual referred to above also contains detailed theory of operation, logic diagrams, interfacing information, operation, maintenance, and wire interconnection data.

A third manual is also available in this series: Programing Manual for PAC-16 Programable Adaptable Controller. This manual contains a description of the PAC-16 Symbolic Assembler together with other information of interest to the Programer.

TABLE OF CONTENTS

Section		Page
1	INTRODUCTION	1-1
1-1	General	1-1
1-2	Specifications	1-1
1-3	PAC-16 Instruction Set	1-3
2	DESCRIPTION	2-1
2-1	General	2-1
2-2	Mnemonic List	2-1
2-3	Block Diagram Discussion	2-5
2-4	Common Bus	2-5
2-5	Common Bus Multiplexer Control	2-7
2-6	Instruction Sequencer	2-7
2-7	Instruction Sequencer Control	2-7
2-8	Program Counter (PC)	2-7
2-9	Program Counter Control	2-8
2-10	Instruction Register	2-8
2-11	Instruction Register Control	2-8
2-12	Instruction Decoder and Control	2-8
2-13	Accumulator	2-9
2-14	Full Adder and Control	2-9
2-15	Accumulator Register and Control	2-9
2-16	Accumulator Multiplexer and Control	2-9
2-17	Memory and Control	2-10
2-18	Manual Controls and Indicators	2-10
2-19	Input-Output	2-10
2-20	Instruction Repertoire Description	2-10
2-21	Fetch Cycle	2-10
2-22	SKP (Skip)	2-12
2-23	STO (Store Accumulator)	2-15



TABLE OF CONTENTS (Continued)

Section	Page
2-24 STA (Store Adder)	2-15
2-25 ADD	2-15
2-26 LDD (Local Double)	2-15
2-27 LDA	2-16
2-28 LA (Load Address)	2-16
2-29 JMP (Jump)	2-16
2-30 MIC (Micro Instruction)	2-16
2-31 IOC (Input-Output Control).	2-16
2-32 JST (Jump and Store)	2-16
2-33 ISK (Increment and Skip)	2-17
2-34 SH (Shift)	2-17
2-35 Pseudo Operations	2-18
2-36 Manual Machine Cycles	2-19
2-37 STRM (Store Manual)	2-19
2-38 FILL (Fill Memory)	2-19
2-39 Theory of Operation	2-20
2-40 Common Bus	2-20
2-41 Common Bus Multiplexer Control	2-21
2-42 Instruction Sequencer and Sequencer Control	2-21
2-43 Program Counter and Control	2-23
2-44 PC Parallel Entry	2-23
2-45 PC Increment Control	2-23
2-46 Instruction Register and Control	2-24
2-47 Instruction Decoder and Control	2-24
2-48 Accumulator	2-25
2-49 Full Adder and Control	2-25
2-50 Accumulator Register and Control	2-25
2-51 Accumulator Multiplexer and Control	2-26



LIST OF TABLES

Table		Page
1-1	Specifications	1-1
2-1	Mnemonic Signal Names and Definitions	2-1



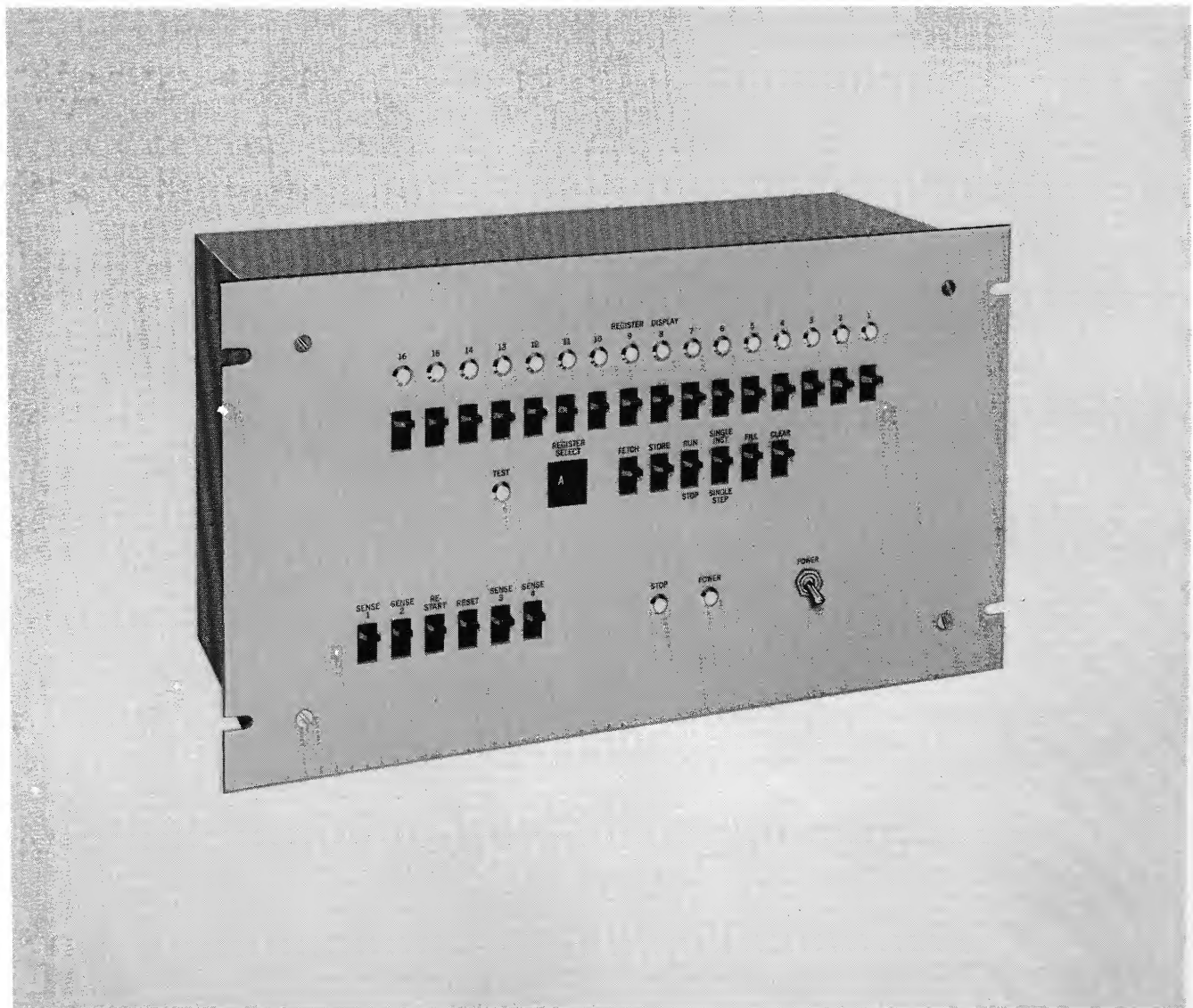


Figure 1-1. PAC-16 Programmable Adaptable Controller





TABLE 1-1. SPECIFICATIONS (Continued)

Circuitry:	MSI-Dual in-line T ² L integrated circuits
Signal Levels:	Logic ZERO = 0V dc Logic ONE = +5V dc
External Power Recommended:	1K memory +5V dc - 6.5 Amperes +12V dc - 2.5 Amperes -5V dc - 0.5 Ampere 4K memory +5V dc = 6.5 Amperes +15V dc - 3.5 Amperes -5V dc - 0.5 Ampere
Environment:	Temperature 0°C to 50°C
Dimensions:	18 x 12 x 7 inches, mountable in standard RETMA 19-inch rack
Software:	PAC assembler - to be run on Univac 1108 computer directly or via remote terminal Maintenance Diagnostics - provides per- formance test and fault isolation when used in conjunction with front panel controls and indicators

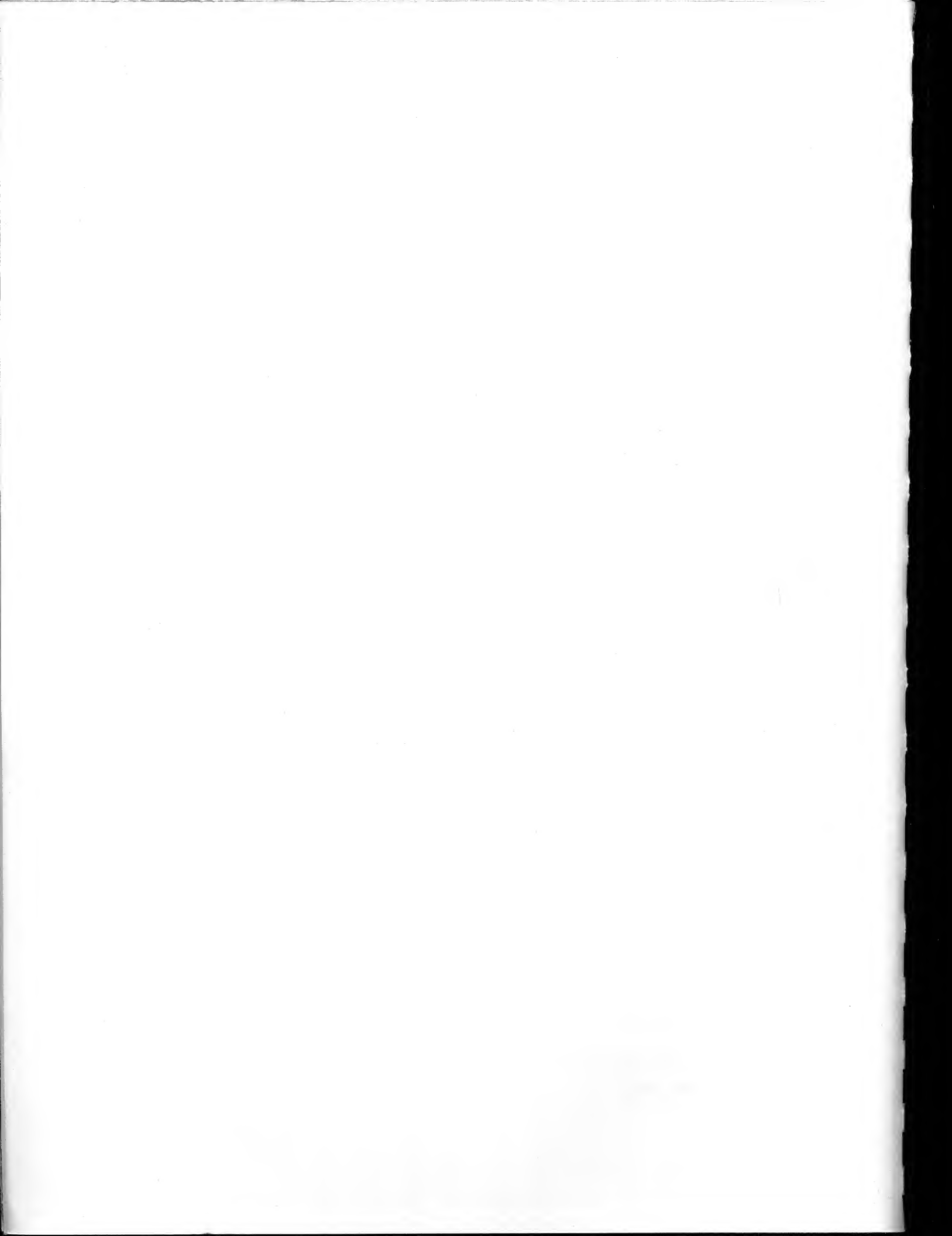


TABLE 1-1. SPECIFICATIONS (Continued)

Circuitry:	MSI-Dual in-line T ² L integrated circuits
Signal Levels:	Logic ZERO = 0V dc Logic ONE = +5V dc
External Power Recommended:	1K memory +5V dc - 6.5 Amperes +12V dc - 2.5 Amperes -5V dc - 0.5 Ampere 4K memory +5V dc = 6.5 Amperes +15V dc - 3.5 Amperes -5V dc - 0.5 Ampere
Environment:	Temperature 0°C to 50°C
Dimensions:	18 x 12 x 7 inches, mountable in standard RETMA 19-inch rack
Software:	PAC assembler - to be run on Univac 1108 computer directly or via remote terminal Maintenance Diagnostics - provides per- formance test and fault isolation when used in conjunction with front panel controls and indicators



1-3. PAC-16 INSTRUCTION SET.

The PAC-16 instructions set is as follows:

PAC-16 INSTRUCTION SET

Mnemonic	Op Code	Exec Time	Summary
Skip A	0	2	Skip next instruction if A is true.
Shift/CMP	1	8	Number of shifts and type of shift or complement in ACC bits 1 to 12.
JST Y	2	7	Store current PC in locations[Y]&[Y + 1]; transfer control to location [Y + 2].
LA Y	3	1	Y_{1-12} into ACC_{1-12} ; bit 12 into ACC_{13-16} .
ISK Y	4	4	[Y] + 1 into [Y]; if [Y] + 1 > 255, skip next instruction.
Spare	5		
LDA Y	6	3	$[Y]_{1-8}$ into ACC_{1-8} .
STA Y	7	6	ACC_{9-12} into $[Y_{1-4}]$ & ACC_{1-8} into $[Y + 1]_{1-8}$.
ADD Y	8	3	$ACC_{1-16} + [Y]_{1-8}$.
MIC A	9	1	Bits 1-12 indicate if MIP is a halt, set/reset interrupt.
JMP Y	A	1	Control is transferred to location Y.
Spare	B		

PAC-16 INSTRUCTION SET (Continued)

Mnemonic	Op Code	Exec Time	Summary
STO Y	C	3	ACC_{1-8} into $[Y_{1-8}]$.
LDD Y	D	6	$[Y]$ into ACC_{9-16} ; $[Y + 1]$ into ACC_{1-8} .
IOC Tx, D	E	4	Tx = any or all of $1 \mu s$ pulses; D is any or all of bits $1-8$ of the IR.
Spare	F		

Instruction Mnemonic	Instruction Format	
	Op	Operand
SKIP	$ \begin{array}{ccccccc} 0 & 0 & 0 & 0 & \overset{-16}{\circ} & \overset{+8}{0} & \overset{1}{1} \\ 16 & & & & 12 & & \end{array} $	$ \begin{array}{ccccccc} \bar{A} & \bar{B} & \bar{0} & \bar{1-8} & \bar{8} & \bar{0} & \bar{1-8} \\ & & & & & & \end{array} $

A binary ONE in any position of bits 1-12 causes the PAC-16 to perform the test indicated - if the condition is met the next instruction (2 bytes) is skipped.

Bit 12 - ($\bar{16}$) Skip if bit 16 = 1 (of ACC)

Bit 11 - ($^{+16}$) Skip if bit 16 = 0 (of ACC)

Bit 10 - (0_{9-16}) Skip if $ACC_{9-16} = 0$

Bit 7 - Skip if $ACC_{1-8} \neq 0$

Bit 6 - ($\bar{8}$) Skip if bit 8 = 1 (of ACC)

Bit 5 - ($^{+8}$) Skip if bit 8 = 0 (of ACC)

Bit 4 - 0_{1-8} Skip if $ACC_{1-8} = 0$

$\bar{A}$ through E are external conditions, if a binary ONE is present the condition will be tested.

PAC-16 INSTRUCTION SET (Continued)

Instruction Mnemonic	Instruction Format
	Op Operand

SKIP (cont)	Extend Mnemonics of skip	Meaning	Hex
	SLNP	$ACC_{1-8} \neq 0$	0028
	SKU	Skip	0038
	NOP	NO-OP	0000
	SKM	$ACC_{16} = 1$	0830
	SKP	$ACC_{16} = 0$	0430
	SKZ	$ACC_{1-16} = 0$	0238
	SKLM	$ACC_8 = 1$	0020
	SKLP	$ACC_8 = 0$	0010
	SKLZ	$ACC_{1-8} = 0$	0008
	SHZ	$ACC_{9-16} = 0$	0230
	SKNA	$A = 0$	0130
	SKNB	$B = 0$	00B0
	SKNC	$C = 1$	0004
	SKND	$D = 1$	0002
	SKNE	$E = 1$	0001

ROR (RIGHT ROTATE)	0	0	0	1	1	1	1	0	X	X	X	X	X	X	X	X
	16				12				8							1

Rotate the ACC right by the amount of ONES present in bits 1-8.

PAC-16 INSTRUCTION SET (Continued)

Instruction Mnemonic	Instruction Format															
	Op				Operand											
RLT (LEFT ROTATE)	0	0	0	1	1	0	1	1	X	X	X	X	X	X	X	X
	16				12				8							1
Rotate the ACC left by the amount of ONES present in bits 1-8.																
NOP (NOP)	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0
No-operation - execution takes 8 cycles.																
(SHIFT RIGHT)	0	0	0	1	1	1	0	0	X	X	X	X	X	X	X	X
	16				12				8							1
Shift the ACC right by the amount of ONES present in bits 1-8. ZEROS are shifted in through the sign bit.																
ALS (SHIFT LEFT)	0	0	0	1	1	0	0	0	X	X	X	X	X	X	X	X
	16				12				8							1
Shift the ACC left by the amount of ONES present in bits 1-8. ZEROS are shifted in the low end.																
CONE (ONES CMP)	0	0	0	1	0	1	0	0	0	0	0	0	0	0	0	1
	16				12				8							1
ONES complement the ACC ₁₋₁₆ .																
CTWO (TWOS CMP)	0	0	0	1	0	1	0	1	0	0	0	0	0	0	0	1
	16				12				8							1

PAC-16 INSTRUCTION SET (Continued)

Instruction Mnemonic	Instruction Format
	Op Operand

TWOS complement the ACC_{1-16} .

JST (JUMP STORE)

$$\begin{array}{ccccccc} 0 & 0 & 1 & 0 & \xleftarrow{\quad [Y] \quad} & & \xrightarrow{\quad} \\ 16 & & & 12 & & & 1 \end{array}$$

Store the contents of the PC counter in the location Y as follows:

PC₈₋₁₂ into [Y]₁₋₄ - DO NOT ALTER [Y]₅₋₈
$$\text{PC}_{1-8} \text{ into } [Y + 1]_{1-8}.$$

LA (LOAD ADDR)

$$\begin{array}{ccccccc} 0 & 0 & 1 & 1 & \xleftarrow{\quad Y \quad} & & \xrightarrow{\quad} \\ 16 & & & 12 & & & 1 \end{array}$$
$$Y_{1-12} \text{ into ACC}_{1-12} \& \text{ACC}_{13-16} = Y_{12} (\text{ACC}_{12})$$

ISK (INCREMENT
SKIP)

0 1 0 0 ← [Y] →

16 12 1

Increment the contents of location $[Y]$ by one and skip next instruction if $[Y] + 1 > 255$.

Spare

0 1 0 1 ← Y → 1
16 12 1

LDA (LOAD ACC)

0 1 1 0 ← [Y] →

16 12

Clears the ACC; goes directly to ADD.

STA (STORE ADDR)

$$\begin{array}{ccccccc} 0 & 1 & 1 & 1 & \longleftarrow [Y] & \longrightarrow & \\ 16 & & & 12 & & & \end{array}$$

Store the ACC_{9-12} into location $[Y]_{1-4}$ & ACC_{1-8}
 $[Y + 1]_{1-8}$.

ADD (ADD)

1 0 0 0 ← [Y] →

Add the contents of Y_{1-8} to ACC_{1-8} .

PAC-16 INSTRUCTION SET (Continued)

Instruction Mnemonic	Instruction Format
	Op Operand

MIC (MICRO INS)

$$\begin{array}{cccccccccccccccc} 1 & 0 & 0 & 1 & X & X & X & X & X & X & X & X & X & X & X & X \\ 16 & & & & 12 & & & & & & & & & & & 1 \end{array}$$

A binary ONE in any position of bits 1-12 causes the following instruction to be executed.

Bit 12 - Halt

Bit 4 - Set interrupt #1 (Enable)

Bit 2 - Set interrupt #2 (Disable)

Bit 3 - Reset interrupt #1 A

Bit 1 - Reset interrupt# 2 B

All bits = 0, NOP is executed (one cycle)

JMP (JUMP)

$$\begin{array}{ccccccc} 1 & 0 & 1 & 0 & \xleftarrow{\quad [Y] \quad} & & \xrightarrow{\quad} \\ 16 & & & 12 & & & 1 \end{array}$$

Control is transferred to the instruction at location Y.

Spare

$$\begin{array}{ccccccc} 1 & 0 & 1 & 1 & \longleftarrow & Y & \longrightarrow \\ 16 & & & 12 & & & 1 \end{array}$$

STO (STORE ACC)

$$\begin{array}{ccccccc} 1 & 1 & 0 & 0 & \xleftarrow{\quad} & [Y] & \xrightarrow{\quad} \\ 16 & & & 12 & & & 1 \end{array}$$

Store the ACC_{1-8} in location Y_{1-8} .

LDD (LOAD DOUBLE)

$$\begin{array}{ccccccc} 1 & 1 & 0 & 1 & \longleftarrow & [Y] & \longrightarrow \\ 16 & & & 12 & & & 1 \end{array}$$

Load the contents of $[Y]$ into ACC_{9-16} and load the contents of $[Y + 1]$ into ACC_{1-8} .

Spare

1 1 0 1 Y
16 12 1

PAC-16 INSTRUCTION SET (Continued)

Instruction Mnemonic	Instruction Format	
	Op	Operand

IOC (INPUT-
OUTPUT)

1 1 1 0 Td Tc Tb Ta ← D →
16 12 8 1

ta } A binary ONE in any or all of
tb } these positions causes a one
tc } microsecond pulse to be output
td } as follows:

1 4 μ S 1

ta NOTE:
tb ta also causes the
tc ACC₁₋₈ = 0.
td

D A binary ONE in any or all of these
positions is available for inspection
through the IR.

Spare

1 1 1 1 ← Y →
16 12 1

PAC-16 INSTRUCTION SET - OPTIONS

Mnemonic	Op Code	Description
CODE Y	5	Count the number of ONES or ZEROS in a string and store count in ACC; load/ store index registers from/into the ACC.
MAP Y	B	Map ONES or ZEROS for the count in the ACC; load/store index registers from/into the ACC.

PAC-16 INSTRUCTION SET - OPTIONS

Mnemonic	Op Code	Description
----------	---------	-------------

SKEX Y	F	Skip next instruction on any one of twelve external conditions as specified by bits 1 through 12.
--------	---	---

Instruction Mnemonic	Instruction Format
----------------------	--------------------

CODE	0	1	0	1	← S →	X	X	X	X	X	X	X	X
	16			12		8							1

S is a four bit sector address which, when added to the index register, forms a memory address; the bit within any byte is held in a three bit register (save bit).

Bit 1 - ONE

Bit 2 - Clear ACC

Bit 3 - 1- Count ONES

0- Count ZEROS

Bit 4 - Single bit elimination

The code instruction begins counting (successive ONES or ZEROS) at the locations and bit specified and places the sum in the ACC.

PAC-16 INSTRUCTION SET - OPTIONS (Continued)

Instruction Mnemonic	Instruction Format
----------------------	--------------------

MAP	$ \begin{array}{cccccccccccccccc} 1 & 0 & 1 & 1 & \leftarrow S \rightarrow & X & X & X & X & X & X & X & X & X \\ 16 & & & 12 & & & 8 & & & & & & & 1 \end{array} $
-----	--

S is a four bit sector address which, when added to the index register (XR), forms a memory address; the bit within any byte in the field is held in save bit (a three bit register).

Bit 1 - 1 for MAP

0 for control

Bit 2 - Clear AC

Bit 3 - 1 Map ONES

0 Map ZEROS

Bit 4 - Not used

Bit 5 - ACC₁₋₃ into save bit

ACC₄₋₁₁ into XR

Bit 6 - Save bit into ACC₁₋₃

XR into ACC₄₋₁₁

Bit 8 - Save bit into ACC₁₋₃

The map instruction begins placing ONES or ZEROS in the memory location and bit specified for the count in ACC.

SKEX (SKIP EXTERNAL)	$ \begin{array}{cccccccccccccccc} 1 & 1 & 1 & 1 & X & X & X & X & X & X & X & X & X & X & X \\ 16 & & & 12 & & & & & & & & & & & 1 \end{array} $
-------------------------	---

If a binary ONE is present, the external conditions associated will be tested; if high, the next instruction will be skipped. [All twelve conditions are logically ORed.]

SECTION 2

DESCRIPTION

2-1. GENERAL.

The PAC-16 Programmable Adaptable Controller is a device capable of controlling one or more external equipments based on a stored program. Wide flexibility is available to the System Designer with respect to input-out capability and the large number of internal functions that can be controlled from an external location.

This section contains mnemonic definitions, a block diagram discussion, instruction repertoire description, and theory of operation.

2-2. MNEMONIC LIST.

Table 2-1 contains a list of mnemonic signal terms and their definitions.

TABLE 2-1. MNEMONIC SIGNAL NAMES AND DEFINITIONS

Mnemonic	Definition
ACCADD	Accumulator Add
ACCADD4	Accumulator Add [A Switch]
ACCADD5	Accumulator Add [Shift Instruction]
ACCCL	Accumulator Clear
ACC1 - ACC16	Accumulator Bits 1-16
ADD	ADD Instruction
ADDA	ADD Instruction [ADD · LOAD]
ADD1 - ADD3	ADD Instruction Cycles
C1 - C3	Carry
CB1 - CB16	Common Bus
CYCLE INITIATE	[Memory] Cycle Initiate
DBI1 - DBI8	Data Bit In [Memory]
DBO1 - DBO8	Data Bit Out [Memory]
DECODE	Decode [Instruction]
$\Sigma 1 - \Sigma 16$	Adder Sums 1-16

TABLE 2-1. MNEMONIC SIGNAL NAMES AND DEFINITIONS (Continued)

Mnemonic	Definition
FETCH	FETCH Instruction
FETCH 1 - FETCH 6	FETCH Instruction Cycles
FETCHM	FETCH Manual
FETCHS	FETCH Switch
FILLS	Fill Switch
FILL 1-4	Fill Control Cycles
FTFF	First Time Flip-flop
HALT	Halt [Clock]
GND E, F, M, N	Ground, Row
GND 1, 5	Ground, Connector
INITIATE READ	Memory Control [1K]
INITIATE READA	Memory Control [4K]
INITIATE WRITE	Memory Control [1K]
INITIATE WRITEA	Memory Control [4K]
IOC	Input-Output Command
IOC1 - IOC4	Input-Output Command Cycles
IRINC	Instruction Register Increment
IRLD4	Instruction Register Load [I Switch]
IRLDLS	Instruction Register Load Least Significant
IRLDMS	Instruction Register Load Most Significant
IR1 - IR16	Instruction Register Bits
ISK	Increment and Skip Instruction
ISK1 - ISK5	Increment and Skip Instruction Cycles
JMP	Jump Instruction
JMP1	Jump Instruction Cycle
JST	Jump Store Instruction
JST1 - JST7	Jump Store Instruction Cycles
LA	Load Accumulator

TABLE 2-1. MNEMONIC SIGNAL NAMES AND DEFINITIONS (Continued)

Mnemonic	Definitions
LA1 - LA2	Load Accumulator [Cycles]
LDD	Load Accumulator Double
LDD1 - LDD3	Load Accumulator Double Cycles
LOAD	Load Instruction
LRB1 - LRB16	Lamp Register Bits
MIC	Micro Instruction
MIC1	Micro Instruction Cycle
MIP and MIP1 (SA)	Memory Input Address
MR	Memory Read
MRMSB	Memory Read Most Significant
MRW	Memory Rewrite
MSTCLK, MSTCLKA	Master Clock
MWRLS	Memory Write Least Significant
MWRMS	Memory Write Most Significant
MX1 - MX16	Multiplexer Bits
N23)	Available Instruction Decoder Taps
N26)	
N29)	
OSC	XTAL Oscillator
PCINC	Program Counter Increment
PCINCA	PCINC [Overflow ISK]
PCINC1	PCINC [Skip Test]
PCLD	Program Counter Load
PCLD4	Program Counter Load [P Switch]
PC1 - PC16	Program Counter Bits
PE	Parallel Entry
PS1 - PS16	Panel Switch Bits
PSSTR	Panel Switch Strobe

TABLE 2-1. MNEMONIC SIGNAL NAMES AND DEFINITIONS (Continued)

Mnemonic	Definitions
RESET	Reset [Switch]
RESET ADDRESS	Reset Memory Address
RESTART	Restart Switch
RESTART FF	Restart Flip-Flop
SA	Switch A Register
SET ADDRESS	Set Memory Address
SH	Shift Instruction
SHC	Shift Control
SHC1-8	Shift Instruction Cycles
SI	Switch I Register
SINGLE INSTRUCTION	-
SINGLE STEP	-
SKP	Skip Instruction
SKP1 - SKP2	Skip Instruction Cycles
SLIN	Shift Left Input
SM	Switch M Register
SP	Switch P Register
SRIN	Shift Right Input
STA	Store Accumulator Instruction
STA1-3	STA Cycles
STO	Store Instruction
STO1 - STO3	STO Cycles
STOP	-
STOP ENABLE	-
STOPFF	Stop Flip-flop
STOP LIGHT	-
STORES	Store Switch
S0	Multiplexer Control

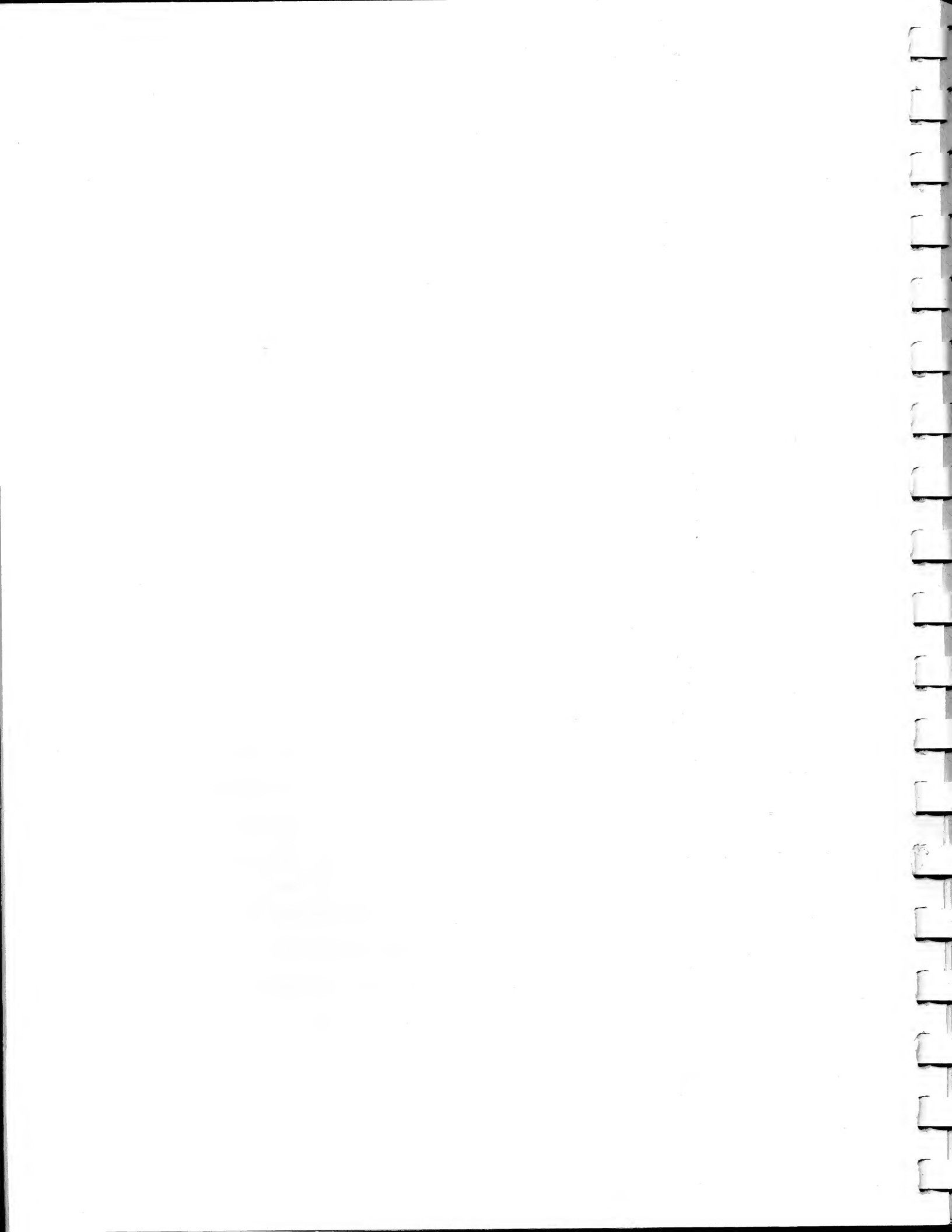
TABLE 2-1. MNEMONIC SIGNAL NAMES AND DEFINITIONS (Continued)

Mnemonic	Definitions
S1	Multiplexer Control
S2	Multiplexer Control
TA	Timing Pulse A
TB	Timing Pulse B
TC	Timing Pulse C
TD	Timing Pulse D
TEST	-
ZEROLS	Accumulator Zero Least Significant
ZEROMS	Accumulator Zero Most Significant
VCCD	VCC Row D
VCC5	VCC Conn. 5

2-3. BLOCK DIAGRAM DISCUSSION. (See figure 2-1.)

The PAC-16 controller is operated by use of front panel controls from an external location based on the use of optional external control functions. The paragraphs which follow describe the controller as operated from the front panel with appropriate notes where external control options are available.

2-4. COMMON BUS. The common bus is the focal point for transfer of all data in the machine (this includes receipt of data from an external source and transmission of controller data to an external destination). The common bus is made up of 12 eight-input multiplexers and 4 four-input multiplexers, a total of 16 stages. Five of the eight common bus channels in stages CBMX1 through CBMX8 are utilized for internal controller signal flow; three channels in these stages are available for system design purposes. Stages CBX9 through CBX12 contain six channels utilized for internal signal flow (two channels are available to the System Designer in these stages) and stages CBMX13 through CBMX16 contain three channels for internal use and one available for system design. Note that additional common bus channels can be added as desired by inserting additional logic elements in the space provided.



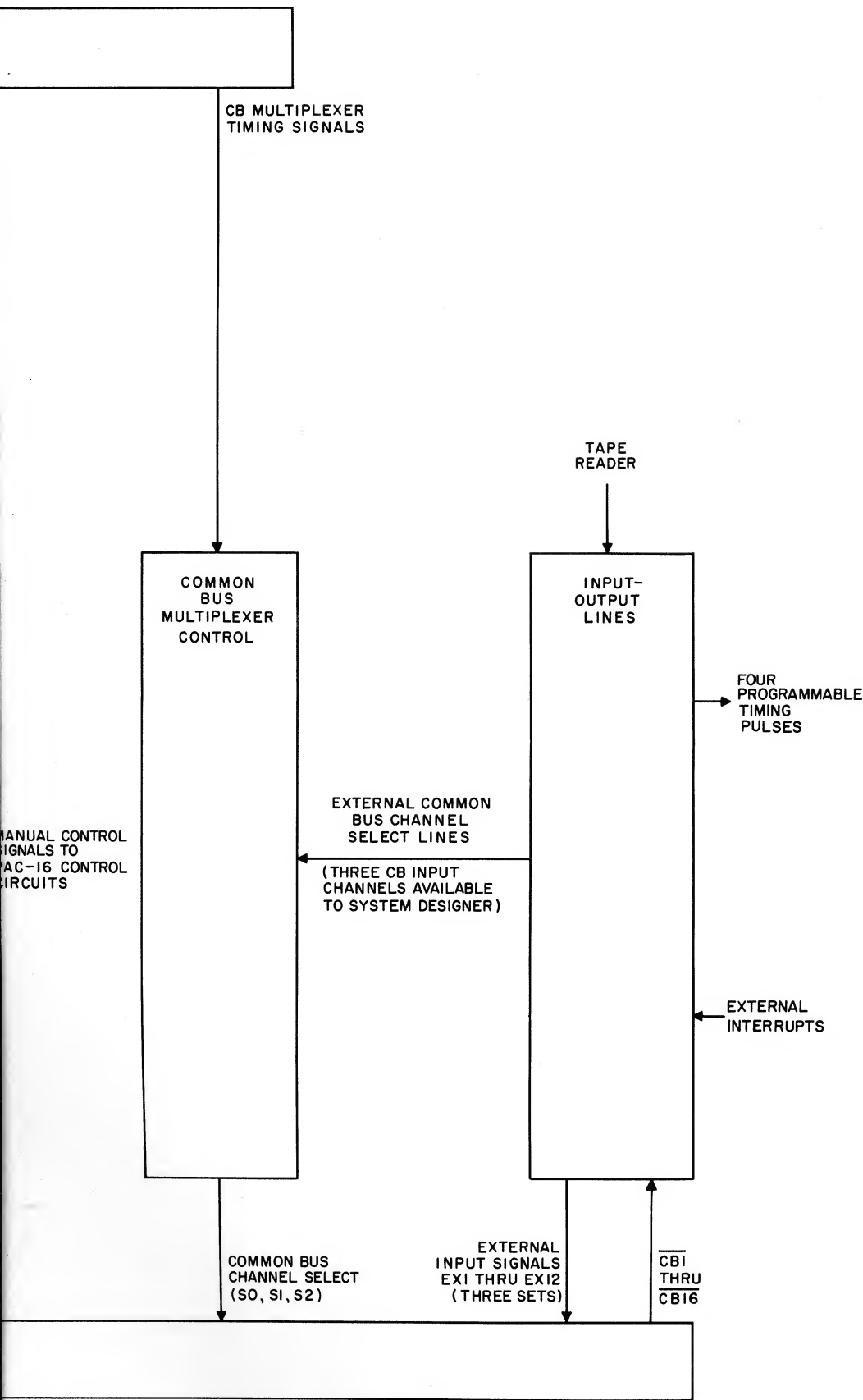
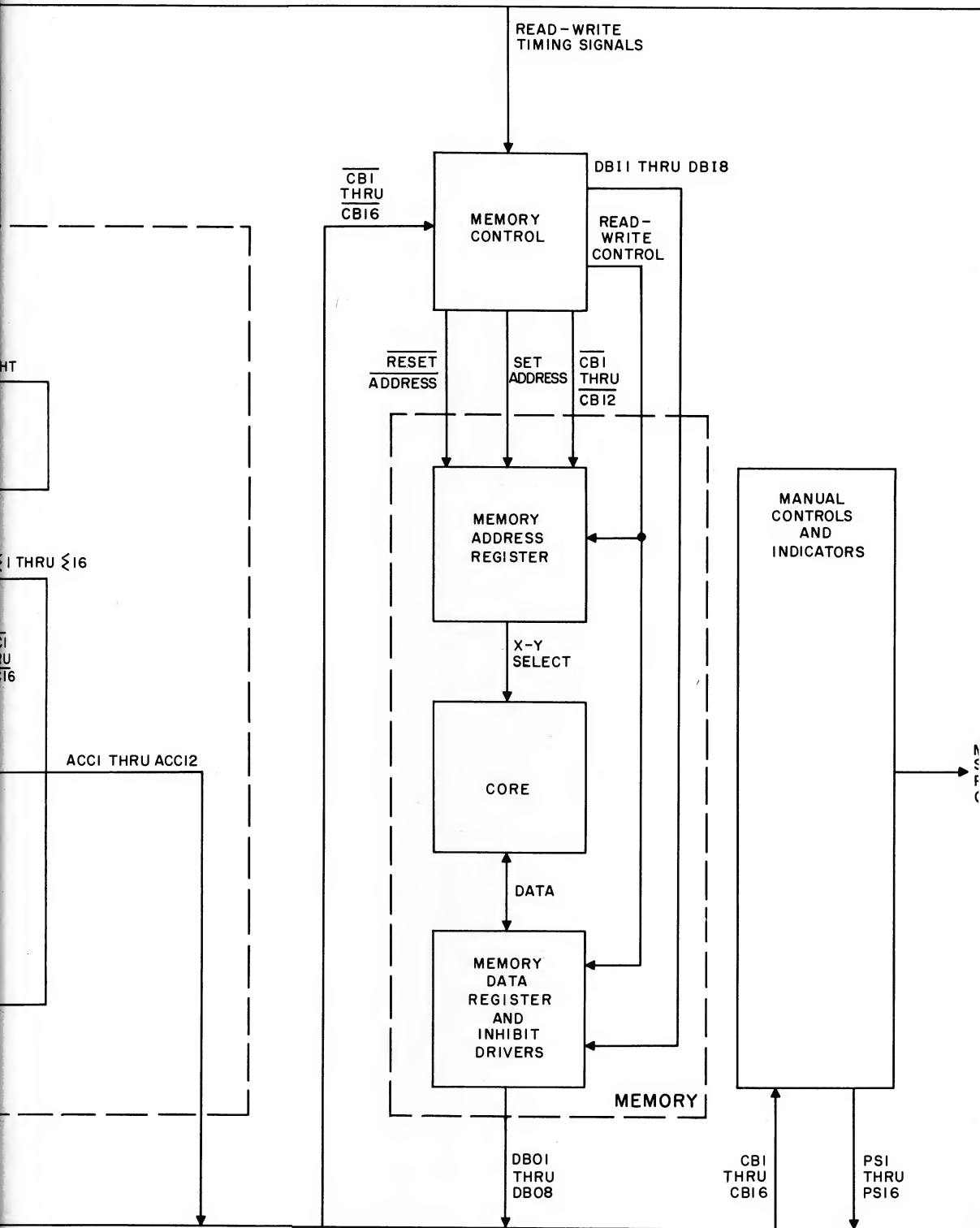
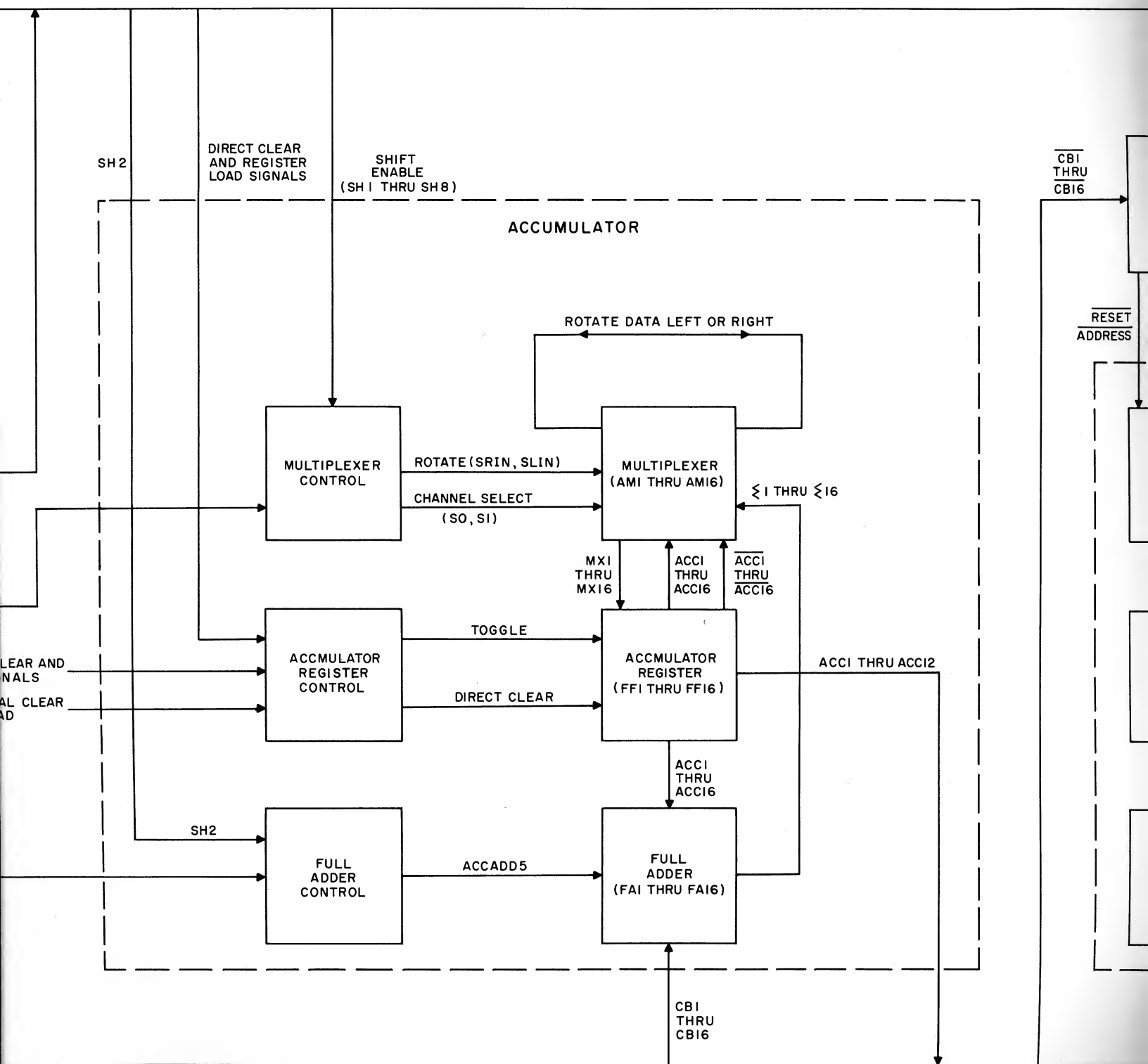


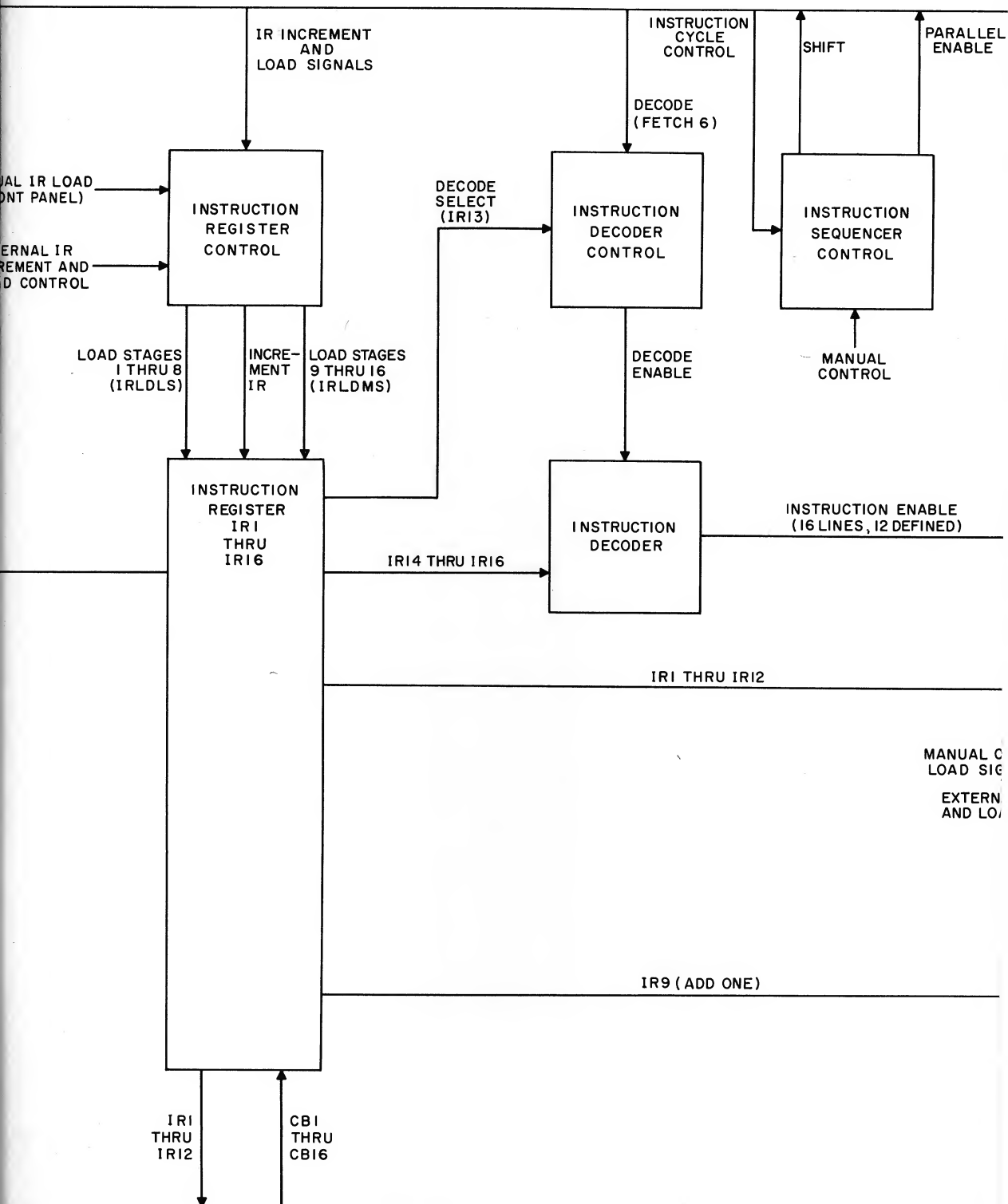
Figure 2-1. PAC-16, Block Diagram

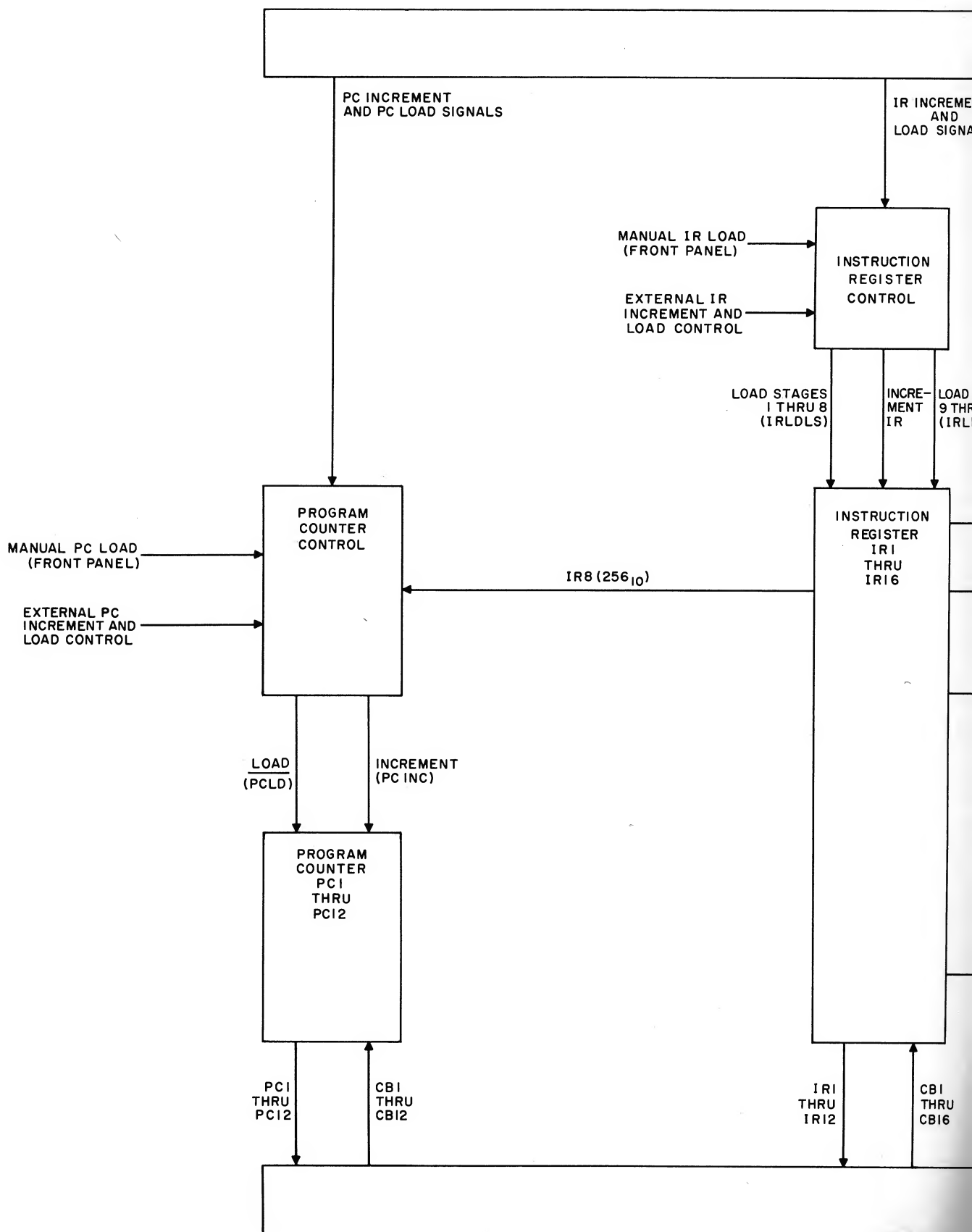


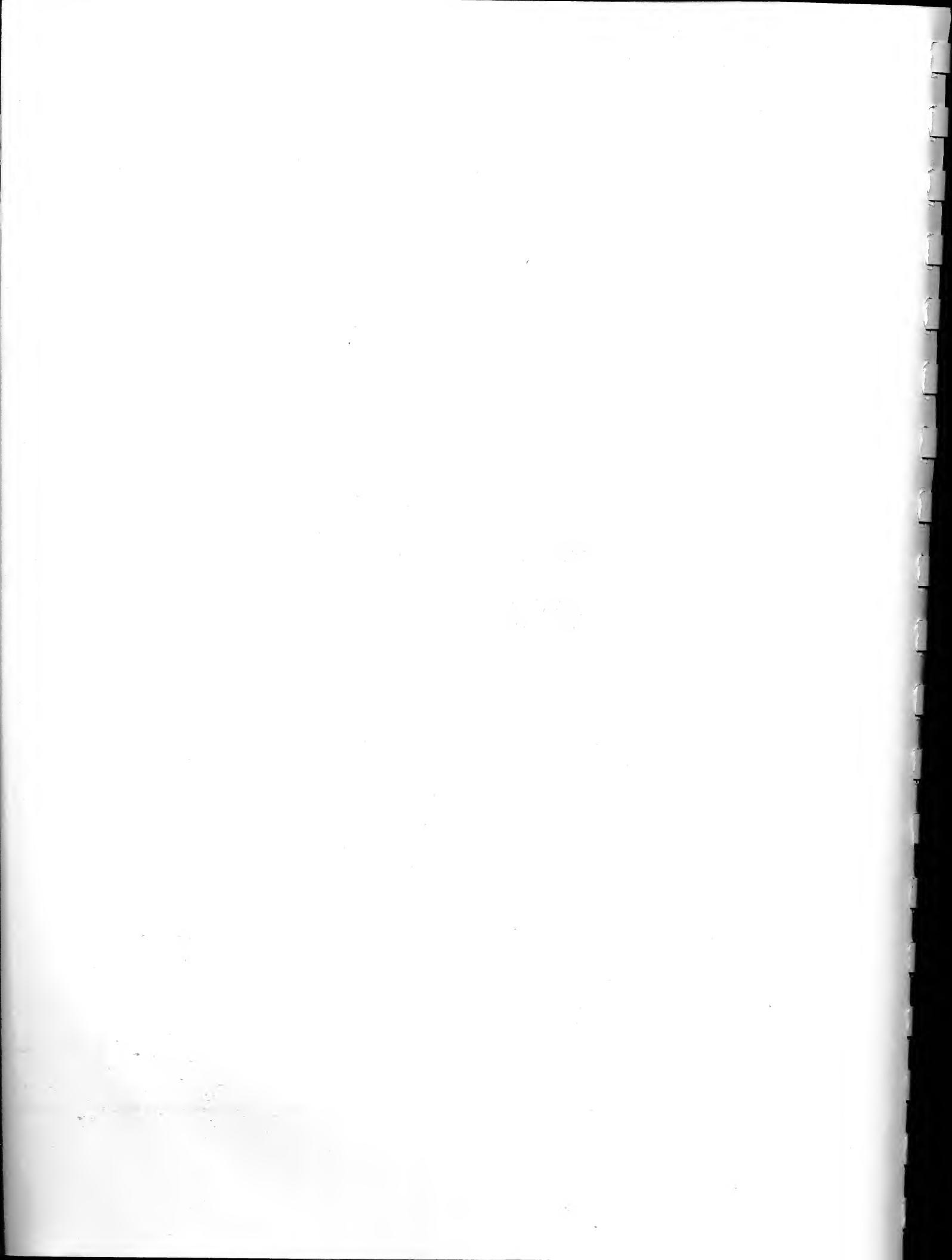
INSTRUCTION SEQUENCER (SHIFT REGISTER MSI'S SRI THRU SRI4)



COMMON BUS MULTIPLEXER (CBMXI THRU CBMXI6)







Common bus channels are selected by a three-bit input code applied to each eight-input multiplexer, and by a two-bit code applied to the four-input multiplexers. An inhibit input is also available which is used to inhibit multiplexer action in stages CBMX1 through CBMX12 for certain machine operations.

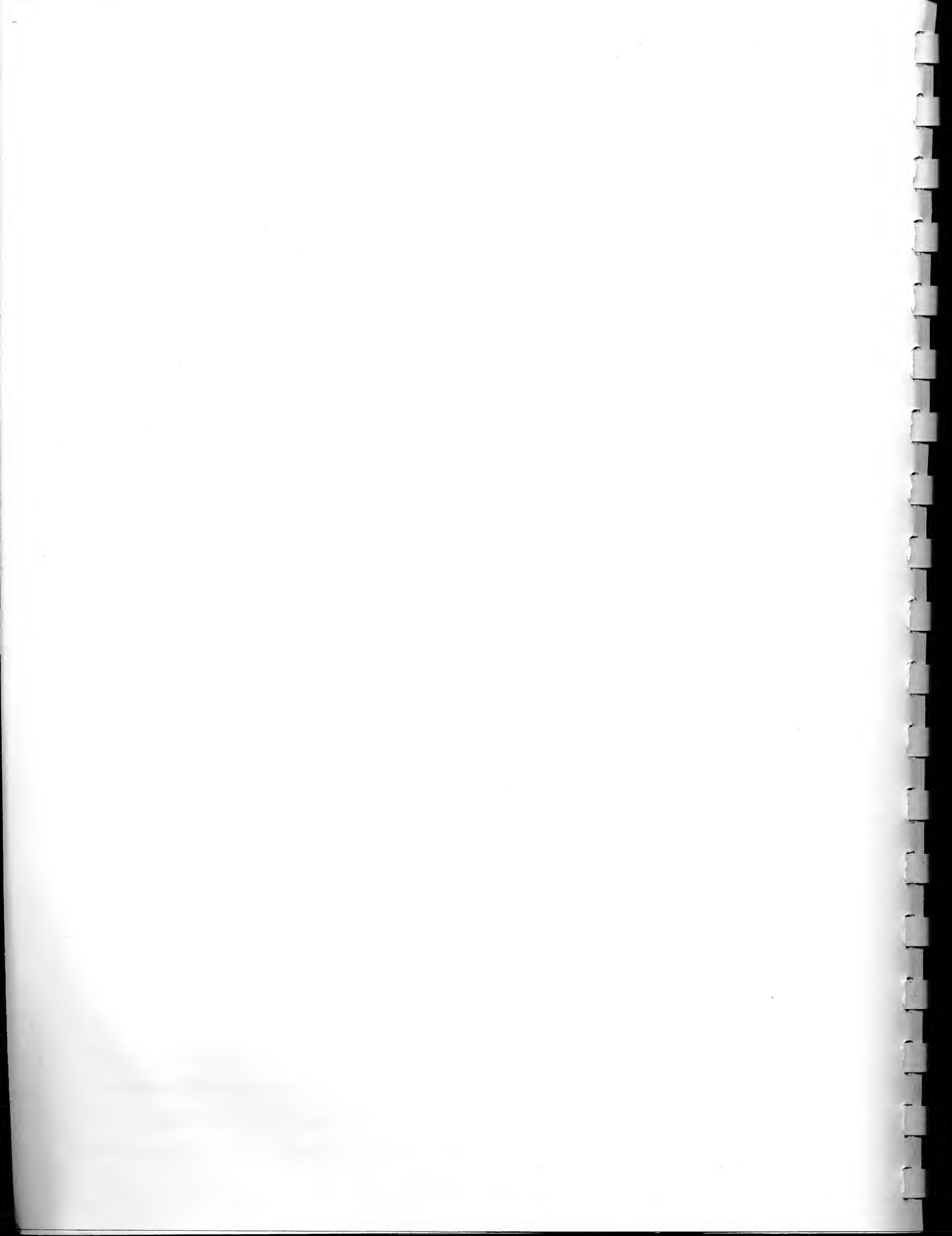
2-5. COMMON BUS MULTIPLEXER CONTROL. Common bus channel selection signals are developed based on inputs from the instruction sequencer during portions of individual instruction cycles. Provision is also included to select channels manually from front panel controls.

2-6. INSTRUCTION SEQUENCER. The instruction sequencer controls all aspects of signal flow in the machine based on application of instruction select signals from the instruction decoder or manual controls. The instruction sequencer consists of 14 four-stage shift register MSI's that are used in this application to accept one instruction enable signal at a time (parallel entry) and then to shift the signal through the number of shift stages associated with the decoded instruction. The shift operation occurs at the clock rate. Some instructions require one clock time; others as many as eight clock times. When an instruction is decoded, the sequencer provides one (and only one) discrete output for each clock phase of an individual instruction cycle. For example, the ADD instruction requires three clock times. Initially, term $\overline{ADD1}$ is applied to stage one of shift register SR3. On the start of the ADD cycle (first clock time) the sequencer generates term $\overline{ADD1}$; during the next clock time $\overline{ADD2}$ is active and during the third clock time $\overline{ADD3}$ is active. When the third clock time is complete, the ADD instruction cycle is over.

The three terms ($\overline{ADD1}$, $\overline{ADD2}$, and $\overline{ADD3}$) generated during the ADD instruction cycle define the operations performed in the machine during that cycle.

2-7. INSTRUCTION SEQUENCER CONTROL. The control circuits for the sequencer consist of gates which generate either parallel entry signals or shift signals. Inputs to the instruction sequencer control circuits include timing signals from the sequencer itself (for example, time to bring up the next instruction), and manual control signals.

2-8. PROGRAM COUNTER (PC). The program counter (PC1 through PC12) consists of 3 four-stage binary counters capable of accepting any 12-bit word (parallel entry) from the common bus and of counting from 0 to 4096 sequentially with application of PC increment signals. Parallel entry is enabled by application of a load signal and the program counter can be cleared by application of a reset signal from the front panel. The 12-bit output from the program counter is applied to the common bus.



2-9. PROGRAM COUNTER CONTROL. The program counter control circuitry receives control signals from the sequencer, from front panel switches, and from external sources. In addition, the counter receives a special increment input (IR8) from the instruction register.

2-10. INSTRUCTION REGISTER. The instruction register (IR) is made up of 4 four-stage binary counters identical to the MSI's used in the program counter. The instruction register is used to store instruction words from memory while operations are performed upon them. The device can be incremented (16 stage counter), can be loaded (parallel entry) in independent bytes of eight bits each, and can be cleared by application of a reset signal from the front panel. IR increment and load signals can be applied from external sources at the discretion of the System Designer.

The instruction register receives 16-bit words from the common bus in eight-bit bytes (one byte at a time). Normally the eight most significant bits are loaded into the instruction register first and the eight least significant bits follow. Output bits IR1 through IR12 are connected to the common bus and are also used elsewhere in the equipment as control signals; bits IR13 through IR16 are applied to the instruction decoder and control circuitry.

2-11. INSTRUCTION REGISTER CONTROL. The instruction register control circuitry functions in a manner similar to PC control circuits. The IR can be incremented by a series of signals resulting from individual instructions or from as many as three external signals selected by the System Designer.

Parallel entry (CB1 through 16) to IR1 through 16 is controlled by gating which responds to either internal or external control signals. The IR is loaded in bytes of eight bits each.

2-12. INSTRUCTION DECODER AND CONTROL. The instruction decoder (ID) consists of two BCD-to-decimal decoders connected to function as octal decoders. Each decoder provides one of eight active outputs in response to a three-bit input code (IR14 through IR16) from the instruction register. One or the other decoder is enabled depending on the state of term IR13. The instruction decode process is initiated by application of a DECODE signal from the instruction sequencer logic.



2-13. ACCUMULATOR. The accumulator is a device which contains a multiplexer, a storage register, a full adder, and control circuitry for each of these sub units. Data are transferred between the common bus and accumulator in both directions. Control signals are applied from the sequencer, the instruction register, and from external sources.

2-14. Full Adder and Control.

The accumulator full adder (FA1 through FA16) is contained in four MSI's and provides 16 outputs to the accumulator multiplexer which are equal to the sum of the contents of the common bus and the accumulator register. Provision is made to insert a ONE in the first adder stage for TWO's complement arithmetic or to add ONE to the accumulator.

2-15. Accumulator Register and Control.

The accumulator register consists of 16 D-type flip-flops used to store the outputs from the accumulator multiplexer. Flip-flop ONE and ZERO outputs (AC1 through 16 and ACC1 through ACC16) are applied to the accumulator multiplexer. Control circuitry provides the means by which the flip-flops are cleared (DIRECT CLEAR) and toggled, based on signals from the sequencer, manual signals from the front panel or external clear and load signals.

2-16. Accumulator Multiplexer and Control.

The accumulator multiplexer (AMX1 through AMX16) is a device which transfers one of four inputs to the output of each stage based on a two-bit code (S0 and S1) from the multiplexer control circuitry; the output lines (MX1 through MX16) are applied to the accumulator register.

The multiplexer control receives terms IR1 through IR12 from the instruction register. These terms select the direction of shift of the contents of the accumulator (right or left), how many shifts (from one to eight) and whether to rotate (right or left). Shift timing is supplied from the sequencer to the multiplexer control circuits in the form of shift enable signals.

2-17. MEMORY AND CONTROL. The PAC-16 memory contains a memory address register, a 1024 (or 4096) eight-bit word core, and a memory data register, inhibit drivers, and other control circuits. The memory is "split" cycle, meaning that separate commands are applied for read and write operation.

During read operation the address of the desired word is applied from the common bus to the memory address register (MAR) via control circuits. The memory address register is initially cleared (RESET ADDRESS), then the address is set (SET ADDRESS). A read control signal is applied and the selected word is applied to the common bus (DB01 through DB08). The word is rewritten when a write control signal is applied together with application of terms DBI1 through DBI8 from the common bus to memory inhibit drivers.

2-18. MANUAL CONTROLS AND INDICATORS. The controller is operated by use of front panel controls and indicators. Controls and indicators and their functions are described in table 4-1.

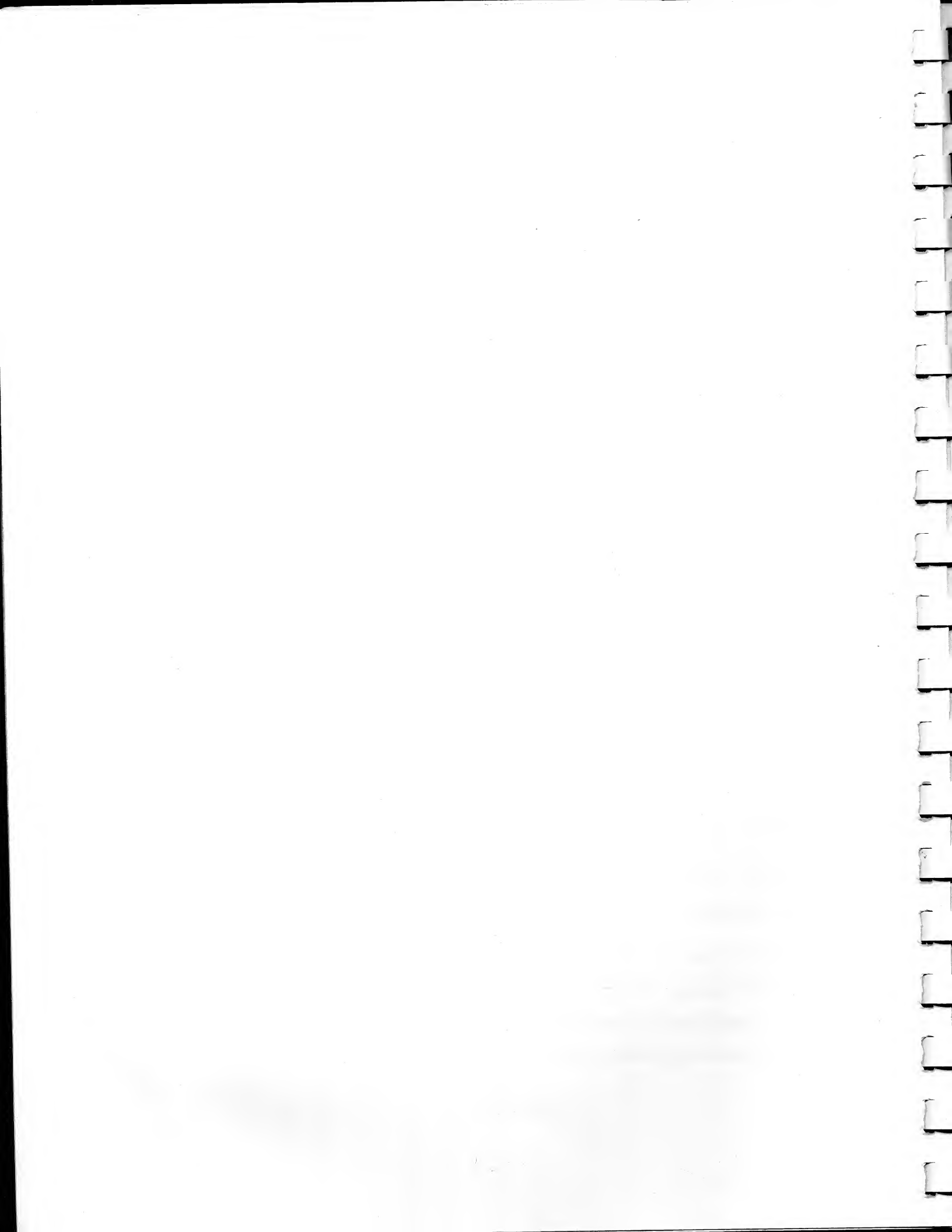
2-19. INPUT-OUTPUT. Provision is included to interface three sets of input lines (eight bits each, plus control lines) with the common bus from an external source. Sixteen common bus lines are available for connection to an external location.

Input terminals for two external interrupt lines are available and four programmable timing pulses are available as outputs from the controller.

2-20. INSTRUCTION REPERTOIRE DESCRIPTION.

The PAC-16 Programmable Adaptable Controller is capable of generating 13 basic timing sequences. The paragraphs which follow describe the timing and movement of data in the machine resulting from the decoding of these sequences (instructions) plus two manual instructions (initiated from the front panel or an external source), and the fetch cycle which is automatic. (See figure 2-2 for overall instruction timing.)

2-21. FETCH CYCLE. The fetch cycle controls the flow of data from memory to the instruction register. It requires six clock times during which memory is addressed and data are transferred and rewritten. The fetch cycle is initiated at the time of completion of each individual instruction or from the front panel (RESET-RESTART).



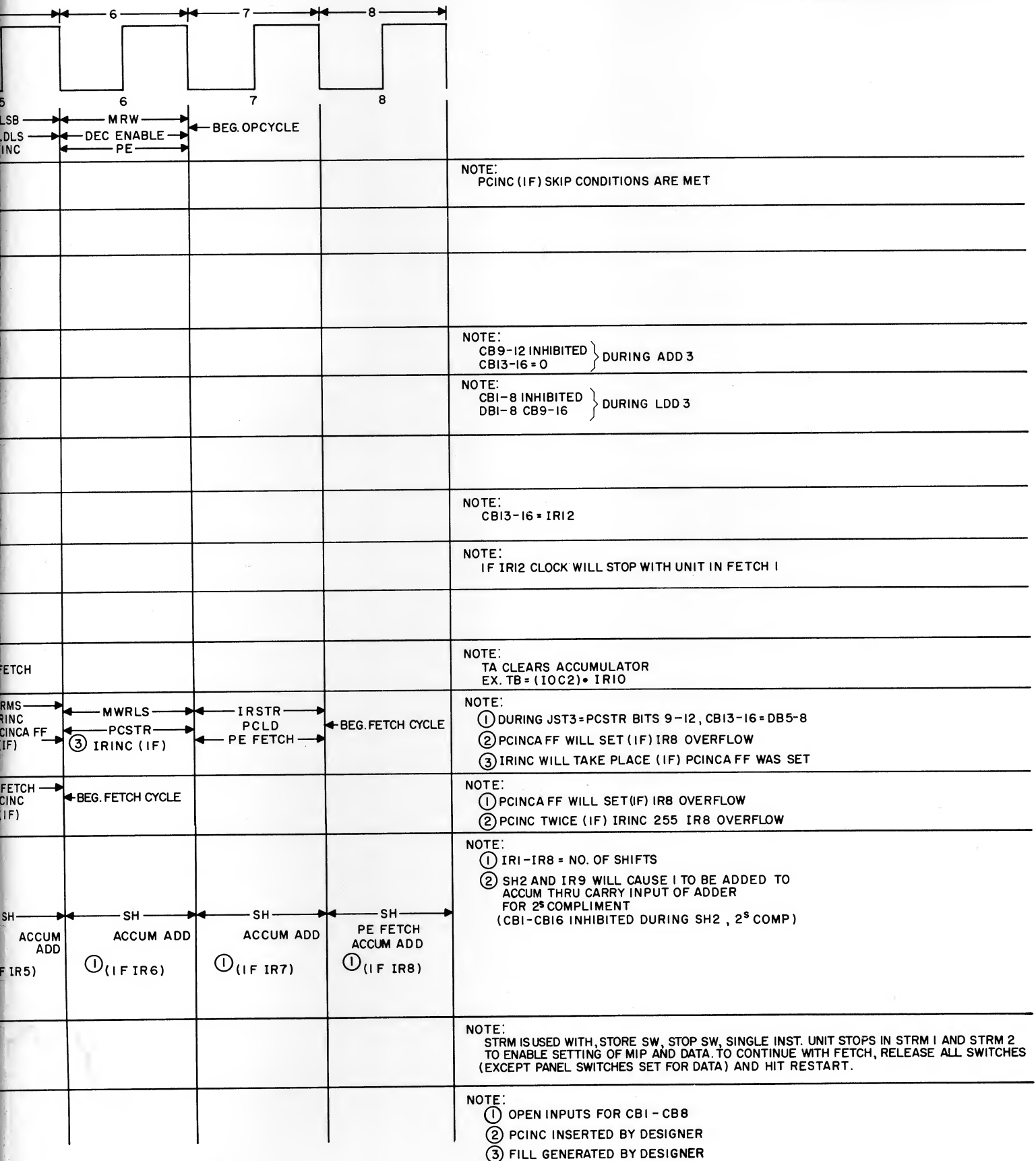
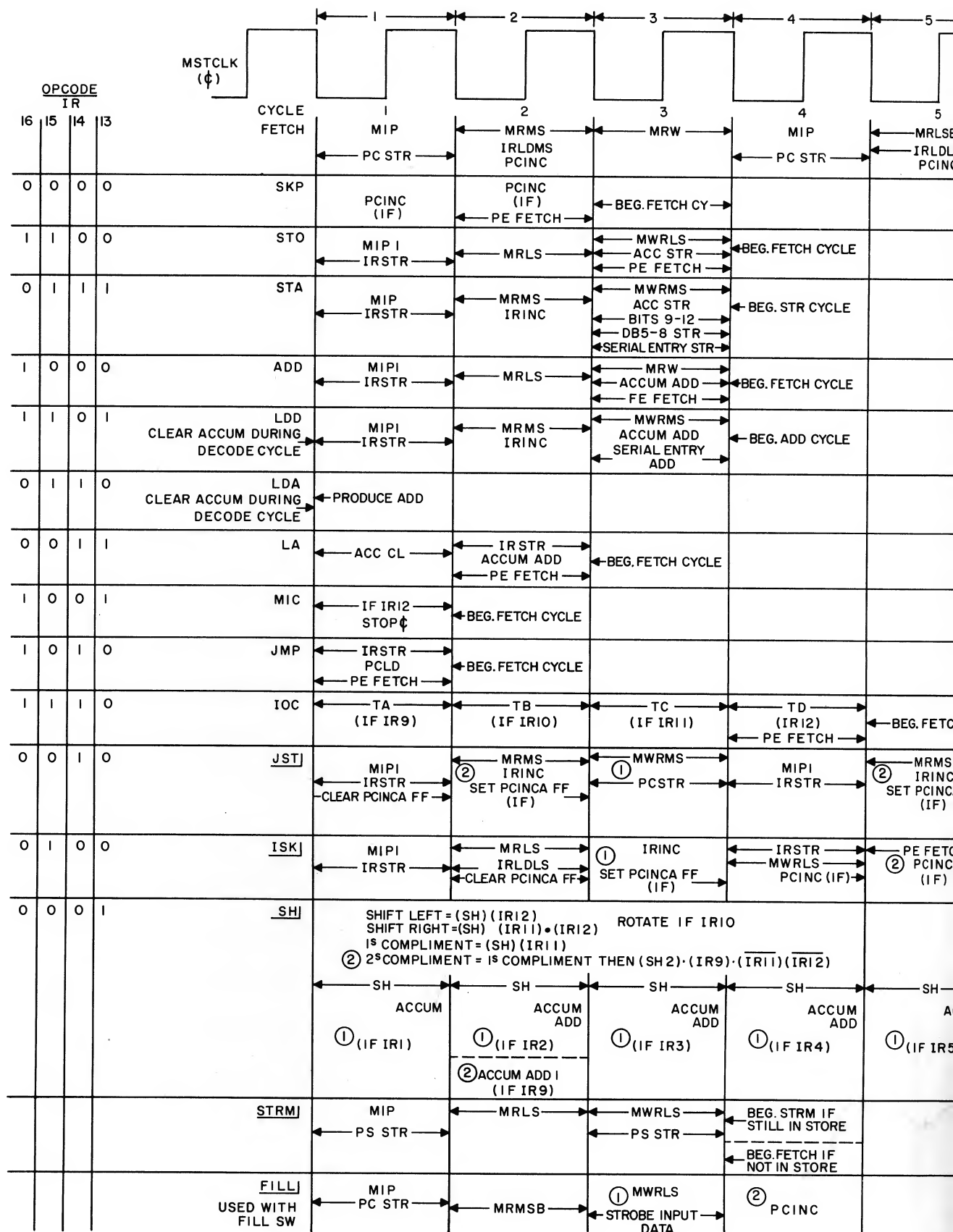


Figure 2-2. Instruction Timing Diagram



- a. Fetch 1 - The contents of the program counter are transferred to the memory address register (MAR) via the common bus. The address is set on the positive-going clock at the mid-point of fetch 1.
- b. Fetch 2 - In fetch 2, the contents of the selected memory location are transferred to the eight most significant stages of the instruction register (IRLDMS). The program counter is incremented by one.
- c. Fetch 3 - During fetch 3, the data are rewritten in memory (same location).
- d. Fetch 4 - The contents of the PC are again transferred to MAR (address following the first address).
- e. Fetch 5 - In fetch 5, contents of the second memory location are transferred to the instruction register (eight least significant bits) and the PC is incremented again. This provides the next memory address (for use when the fetch cycle is completed if a jump, jump store or skip instruction is not decoded).
- f. Fetch 6 - In fetch 6, the word extracted in fetch 5 is rewritten; the instruction decoder is enabled (at this point the instruction register contains 16 bits), and a single active output signal from the decoder is parallel-loaded into the instruction sequencer. From this point, the decoded instruction is executed and a new fetch cycle is automatically initiated.

2-22. SKP (SKIP). The SKP instruction (figure 2-3) tests the conditions under which the instruction which follows (next two memory locations) is skipped or not. In general, a ONE in any of instruction register stages (IR1 through IR12) causes an associated condition to be tested. If the condition is satisfied, the next instruction is skipped; if not, it is performed.

Following decode of the SKP instruction, the sequencer enters SKP1. The following tests are performed simultaneously. All tests must be satisfied in order to skip.

- a. If IR12 is HIGH, is $\overline{\text{ACC16}} = \text{LOW}$? If so skip.
- b. If IR11 is HIGH, is $\text{ACC16} = \text{LOW}$? If so skip.
- c. If IR10 is HIGH, is $\overline{\text{ZEROMS}} = \text{LOW}$? If so skip.

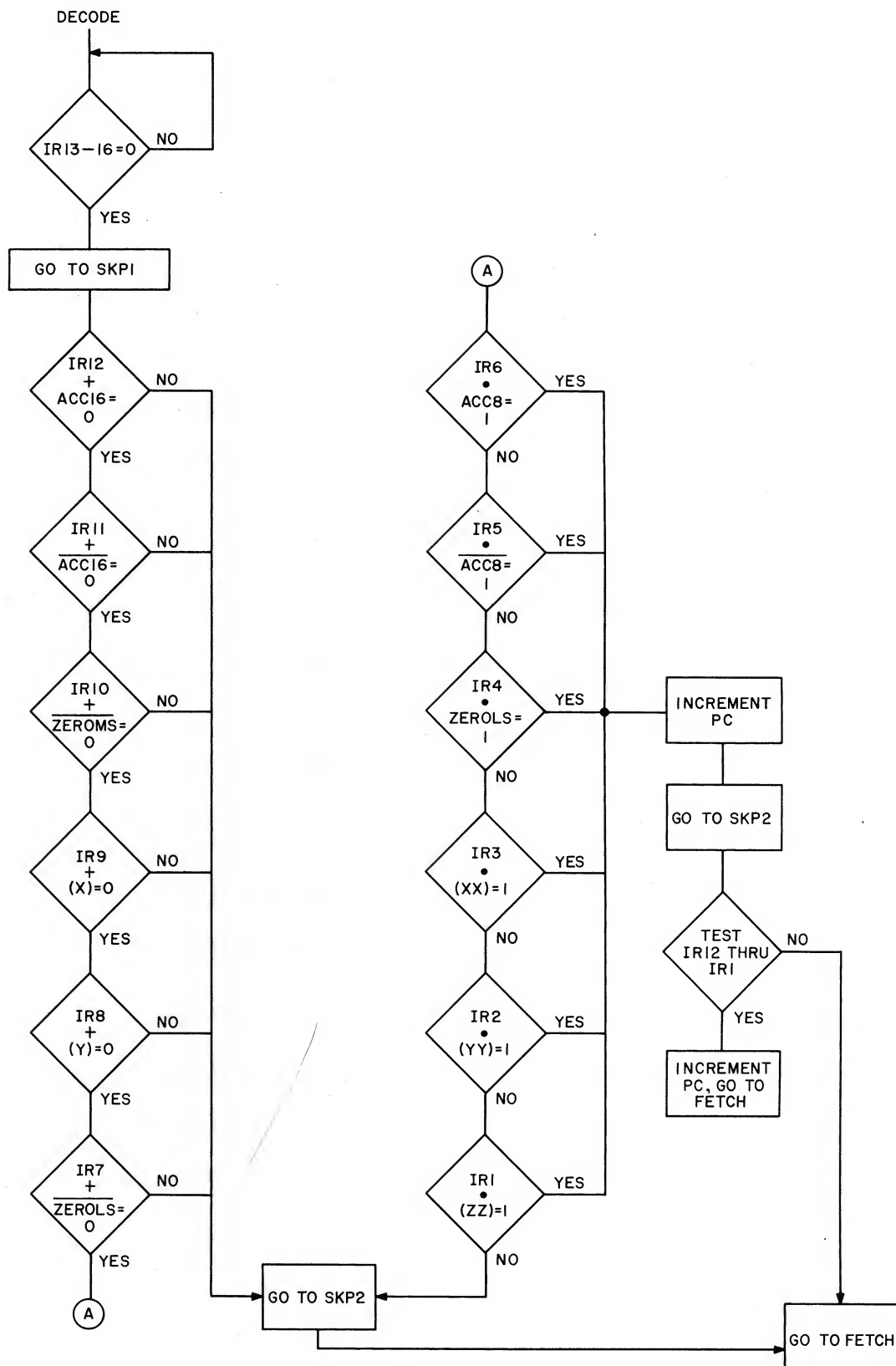


Figure 2-3. SKP Instruction, Flow Diagram

NOTE: Term $\overline{\text{ZEROMS}}$ is active (LOW) if stages 9 through 16 of the accumulator contain ZEROs. Accordingly, this test determines that no ONE is present in any of these stages in order to skip.

- d. IR9 - Test of external condition (X) available to System Designer.
- e. IR8 - Test of external condition (Y) available to System Designer.
- f. If IR7 is HIGH, is $\overline{\text{ZEROLS}}$ LOW? If so, skip.

NOTE: Term $\overline{\text{ZEROLS}}$ is active (LOW) if accumulator stages 1 through 8 contain ZEROs; accordingly, this test determines that no ONE is present in any of these stages in order to skip.

- g. The next test establishes that the sequencer is in either the SKP1 or SKP2 phase of the SKP cycle in order to skip.

In addition to the tests of parts a through g, one of the following sets of conditions must be satisfied in order to skip:

- h. Terms IR6 and ACC8 are HIGH (OR),
- i. Terms IR5 and $\overline{\text{ACC8}}$ are HIGH (OR),
- j. Terms IR4 and ZEROLS are HIGH (OR),
- k. Terms IR3 and (XX) are HIGH (OR),
- l. Terms IR2 and (YY) are HIGH (OR)
- m. Terms IR1 and (ZZ) are HIGH.

NOTE: (XX), (YY) and (ZZ) are external conditions available to System Designer.

Tests a through g must all be satisfied (logic AND of ORs) and one of tests h through m (logic OR of ANDs) must be satisfied in order to skip. If skip conditions are met, the PC is incremented once in SKP1 and again in SKP2. The sequencer then initiates another fetch cycle.

2-23. STO (STORE ACCUMULATOR). The STO instruction requires three clock times. In STO1, the contents of the instruction register are transferred (via the common bus) to the memory address register. In STO2, the memory data register is cleared, and in STO3 the data contained in accumulator stages 1 through 8 are transferred to the memory data register via the common bus. During STO3 conditions are established for initiation of the next fetch cycle with the next clock pulse.

2-24. STA (STORE ADDER). The STA instruction requires six clock times. In STA1 the contents of the instruction register are transferred to the memory address register. In STA2 the instruction register is incremented by one count. (This number remains in the IR for use in the STO cycle which follows STA3.)

During STA3 the contents of accumulator bits 9 through 12 are stored in bits 1 through 4 of the location selected in STA1. In addition, a serial entry enable signal is generated which causes the sequencer to enter STO1. A complete STO cycle is performed during which accumulator bits 1 through 8 are stored in the memory location selected in STA2. The STA instruction occupies three clock times plus the three STO clock times which follow automatically.

2-25. ADD. When the ADD instruction is decoded the contents of IR stages 1 through 12 are transferred to the memory address register (MAR) during ADD1. In ADD2 the contents of the selected memory location are loaded into the memory data register (MDR). In ADD3 the data word is added to the contents of accumulator stages 1 through 8 and stored in the accumulator register. During ADD3 the next fetch cycle is also enabled.

2-26. LDD (LOAD DOUBLE). The LDD instruction transfers the contents of two successive memory locations to the accumulator. Prior to entry into the LDD cycle (during instruction decode) the accumulator is cleared. In LDD1, IR bits 1 through 12 are transferred to MAR. In LDD2 the contents of the selected location are read into MDR and the IR is incremented by one count. In LDD3 the data are added to accumulator stages 9 through 16 (since the accumulator was cleared initially this constitutes a direct accumulator load). In LDD3 a serial entry enable is also set up and the sequencer enters the ADD cycle automatically. During the three clock phases of the ADD cycle the contents of the second memory location (developed in LDD2) are transferred to stages 1 through 8 of the accumulator.

2-27. LDA. The LDA instruction clears the accumulator during the decode cycle (similar to LDD) then initiates the ADD cycle.

2-28. LA (LOAD ADDRESS). The LA instruction clears the accumulator during LA1. In LA2 the contents of the instruction register (bits 1 through 12) are transferred to the accumulator and accumulator stages 13 through 16 are set to the value contained in IR12. In addition, the next fetch cycle is set up.

2-29. JMP (JUMP). The JMP instruction occupies one clock time (JMP1). During this time the contents of the instruction register are loaded into stages 1 through 12 of the program counter. This provides the address of entry into the next fetch cycle which is also set up in JMP1.

2-30. MIC (MICRO INSTRUCTION). When the MIC instruction is decoded, the controller clock is stopped if IR12 = ONE; in addition, the next fetch cycle is set up.

2-31. IOC (INPUT-OUTPUT CONTROL). The IOC instruction results in generation of from zero to four timing pulses designated TA, TB, TC and TD for use externally. During the IOC cycle the following timing applies:

IOC1	IOC2	IOC3	IOC4
IOC · IR9 = TA	IOC · IR10 = TB	IOC · IR11 = TC	IOC · IR12 = TD

Term $\overline{TA}$, when LOW, (TA HIGH) clears the accumulator and IOC4 sets up the next fetch cycle.

2-32. JST (JUMP AND STORE). During JST1 the contents of the instruction register are transferred to MAR and the PCINCA flip-flop is cleared.

NOTE: The PCINCA flip-flop provides a HIGH level at the "1" output when term IR8 goes from HIGH to LOW (count of 256 in the IR). This level is used to increment IR stage 9 during JST6 if the least significant stages (IR1 through IR8) of the IR overflow during JST1.

In JST2 the instruction register is incremented by one and flip-flop PCINCA is set if IR count exceeds 256. During JST3 the content of PC8 through 12 are stored in bits 1 through 4 at the address set in MAR during JST1 (bits 5 through 8 at the location are rewritten without change).

In JST4 the contents of the IR are transferred to MAR. In JST5 the instruction register is incremented by one and flip-flop PCINCA is set if IR stage 8 overflows.

In JST6 the contents of program counter stages 1 through 8 are loaded into the location selected in JST4. Stage 9 of the instruction register is incremented if IR stage 8 overflowed during JST5.

During JST7 the contents of the IR are loaded into the PC and the next fetch cycle is set up. On the trailing edge of the following positive clock fetch starts at the location contained in the PC.

2-33. ISK (INCREMENT AND SKIP). The ISK instruction requires five clock times. During ISK1, the contents of PC are loaded into MAR. In ISK2 the contents of the location selected in ISK1 are transferred to the eight least significant stages of the instruction register; the PCINCA flip-flop is cleared.

In ISK3, the instruction register is incremented by one (PCINCA flip-flop set if IR8 overflows).

During ISK4 the program counter is incremented by one if stage 8 of the IR overflowed during ISK3 and the contents of the IR (bits IR1 through IR8) are written into the location selected in ISK1.

During ISK5 the next fetch cycle is set up and the PC is incremented again if it was incremented in ISK4.

2-34. SH (SHIFT). The SH instruction (OP code 0001) is divided into six categories as defined by bits 9 through 12 of the instruction register. The shift operations are performed on the contents of the accumulator.

a. SRT (shift right) - Shift right is initiated by the presence of a binary ONE in each of stages IR11 and IR12. The number of shifts (up to eight) is selected by stages one through

eight of the instruction register. A binary ZERO is entered into the most significant stage (sign bit) for each programed shift.

b. ROR (right rotate) - Right rotate is initiated by the presence of binary ONES in IR stages 9 through 12. The content of the least significant stage of the accumulator is transferred to the most significant stage during each shift; the number of shifts is determined by the number of ONES in IR stages one through eight.

c. ALS (shift left) - Shift left is initiated by the presence of a binary ONE in IR stage 12. ZEROs are shifted into the least significant stage of the accumulator; the number of shifts is determined by the number of binary ONES in IR stages one through eight.

d. RLT (left rotate) - Left rotate is initiated by the presence of binary ONES in IR stages 9, 10, and 12. The content of the most significant stage is transferred to the least significant stage during each left shift. The number of rotate/shift operations is determined by the number of binary ONES in IR stages one through eight.

e. CONE (ONES complement). The contents of accumulator stages 1 through 16 are complemented when a binary ONE is present in IR stages 1 and 11. Term IR1 enables the accumulator multiplexer during SH1; term IR11 selects the multiplexer code for complement operation.

f. CTWO (TWOs complement). The TWOs complement process is as follows: During SH1, a ONES complement operation is performed on the contents of the accumulator. This requires presence of binary ONES in IR stages 1 and 11. In SH2 a binary ONE in IR stage 9 causes a ONE to be added to the least significant stage of the accumulator through the accumulator full adder.

2-35. PSEUDO OPERATIONS. Two pseudo operations, designated INCA and DECA, are available for incrementing and decrementing the accumulator. These operations make use of the basic SH instruction.

a. INCA (increment accumulator) - To increment the accumulator, a ONE is placed in IR stages 2 and 9. The SH instruction (op code 0001) is programed. During SH2 a binary ONE is applied to the carry input of the first full adder with the result that the accumulator is incremented by one; the result is stored in the accumulator register.

b. DECA (decrement accumulator) - To decrement the accumulator the SH instruction is programmed together with a ONE in stages 1, 3, 9, and 11 of the instruction register. In SH1 a ONEs complement of the contents of the accumulator is performed followed by addition of a ONE during SH2. The accumulator is complemented again during SH3 with the result the one is subtracted from the original value in the accumulator.

2-36. MANUAL MACHINE CYCLES. The following semi-automatic operations are initiated by front panel controls or in conjunction with external control and data sources:

2-37. STRM (Store Manual).

The STRM cycle is initiated by setting REGISTER SELECT to M; STORE switch depressed; RUN-STOP switch in STOP and setting memory address in panel switches PS1 through PS8. RESET switch is depressed and released and RESTART switch is depressed and released. In STRM1, the panel switch address is transferred to MAR and the unit stops. The desired data are entered into the panel switches and RESTART is depressed. Contents of the panel switches are loaded into the location selected in STRM1 and the unit stops. If additional data are to be written into memory the panel switches and RESTART switch are used as in STRM1 and STRM3; if STRM operation is complete, reposition STORE switch and press RESET.

2-38. FILL (Fill Memory).

The fill function is defined by the System Designer depending on the type of interface required for specific applications. Provision is included for application of a LOW level to sequencer MSI SR11, pin 1 (figure 5-4), or to pin 2, in order to initiate four phases of the fill operation (FILL1 through FILL4). A parallel entry enable signal (LOW) must also be applied to gate GR4 at one of the input pins available to the System Designer.

During FILL1, the contents of the program counter are transferred to MAR. In FILL2, a memory read phase is enabled (MRMSB). During FILL3, the common bus is strobed (external input) and the data are transferred to the location selected in FILL1. During FILL4, the PC is incremented based on a signal supplied by the System Designer.

2-39. DESCRIPTION OF CONTROLS AND INDICATORS.

The following listing describes front panel controls and indicators:

Panel Switches 1-16 - Select inputs to the common bus multiplexer to be used either as addresses or as data bits, depending on the sequence being performed.

RESET switch - Clears the controller.

RESTART switch - Starts the Master Clock.

REGISTER SELECT switch - Depending on the operation being performed, this switch directs the store and fetch activities to the proper register.

STORE switch - Allows the Operator to address and alter any location in memory.

FETCH switch - Allows the Operator to retrieve an instruction or instructions from memory and to display the instruction on indicators DS1 through DS16 without executing the instruction.

FILL switch - Prepares the controller to receive DATA from an external device (that is, reader) and stores the data sequentially in memory.

RUN/STOP switch - Normally used in the STOP position when the STORE or FETCH switch is selected. It stops the controller at key cycles of operations as determined by the selection of the above switches.

SINGLE STEP/SINGLE INST. switch - Always used in conjunction with the RUN/STOP switch. When both are selected, the controller cycles at a rate of one clock pulse per depression of the RESTART switch.

SENSE 2 switch - When depressed, the contents of the accumulator are displayed momentarily. The Controller, however, must be stopped, and in a cycle during which an address is displayed.

NOTE: Several switches exist on the front panel which are not included in the above description. They are not connected in the basic controller but are available for system design purposes.

2-40. PROCEDURE FOR FETCHING INSTRUCTIONS.

Proceed as follows:

Step 1. Set front panel switches as follows:

RUN/STOP switch	-	STOP
FETCH switch	-	Depressed
REGISTER SELECT switch	-	M

Set start address in panel switches PS1 through PS12.

Step 2. Depress and release RESET switch; depress RESTART switch once.

Panel lights 1 through 8 will display contents of address.

Step 3. Set REGISTER SELECT switch to I.

Step 4. DEPRESS RESTART switch. Contents of address +1 will be displayed.

Step 5. Further depressions of RESTART switch result in display of the contents of memory at each address. (Address is incremented once per depression.)

2-41. PROCEDURE FOR MANUALLY STORING AN INSTRUCTION.

Set front panel switches as follows:

RUN/STOP switch	-	STOP
STORE switch	-	Depressed
REGISTER SELECT switch	-	M

Step 1. Depress and release RESET switch; set the panel switches to positions corresponding to the address desired.

Step 2. Depress RESTART switch once. The address is now set and panel lights 1 through 8 display the previous contents of that location.

Step 3. Alter the selection of panel switches 1 through 8 to correspond with the data to be inserted.

Step 4. Depress RESTART once. The new data are written into memory at the selected location.

